

Notice of Allowability

Application No.

09/855,386

Examiner

Wei Y. Zhen

Applicant(s)

SPRING, MAXIMILIAN J.

Art Unit

2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to amendment filed on 3/17/2005.
2. ☒ The allowed claim(s) is/are 1,3-8 and 10-29.
3. ☒ The drawings filed on 14 May 2001 are accepted by the Examiner.
4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) ☐ All b) ☐ Some* c) ☐ None of the:
 1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).
 - * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.

THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
 6. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
 - (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
 - 1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
 - (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.
- Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
7. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO-1449 or PTO/SB/08),
Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit
of Biological Material
5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☒ Interview Summary (PTO-413),
Paper No./Mail Date 6/7/05
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____

Art Unit: 2191

1. This office action is in response to the RCE filed on 3/17/2005.
2. Claims 1, 3-8, 10-29 are allowed.

EXAMINER'S AMENDMENT

3. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Christopher J. Palermo, Reg. No.42,056 on 6/7/2005. The amendment was made in order to clarify allowable subject matter in independent claim over the prior arts and in order to overcome a potential 35 U.S.C. 101 rejection.

The application has been amended as follows:

In the Claim

Please cancel claims 2 and 9.

Please amend the claims as followed:

1. (Currently Amended) A method of maintaining version compatibility between a first computer program module and one or more interacting computer program modules that interact with the first module through a interface with capabilities shared by all the interacting modules,

Art Unit: 2191

wherein the modules are stored in computer storage, the method comprising the computer-implemented steps of:

creating first information describing the computer program module interface capabilities at one or more times;

storing the first information in a corresponding plurality of instances of a data structure wherein each instance of the data structure corresponds to the interface capabilities at one or more plurality of times;

creating and storing a mapping that associates the plurality of instances with a corresponding plurality of version numbers for the first computer program module;

automatically assigning a second version number for a second computer program module of the one or more interacting modules based on a corresponding instance from the plurality of instances of the data structure contained in the mapping and based on second information describing interface capabilities of the second computer program module;

determining compatibility of the modules based on a first version number for the first module and the second version number for the second module;

wherein the step of automatically assigning comprises:

describing a subset of the interface capabilities, which subset is employed by the second computer program module;

determining from the plurality of instances at least one instance including data describing the subset of the interface capabilities; and

assigning the second version number for the second module based on the corresponding version number in the mapping and the at least one instance.

2. (Canceled)

3. (Currently Amended) A method as recited in Claim [[2]] 1, wherein said step of assigning the second version number comprises assigning as the second version number a

Art Unit: 2191

particular value of the plurality of version numbers for the first module, the particular value associated with a particular instance of the at least one instance, the particular instance corresponding to an earliest time of one or more times corresponding to the at least one instance.

4. (Previously Presented) A method as recited in Claim 1, wherein the plurality of version numbers for the first computer program module corresponding to the plurality of instances vary in one direction with time of the plurality of times corresponding to the plurality of instances.

5. (Previously Presented) A method as recited in Claim 1, wherein said step of describing the computer program module interface capabilities includes generating and storing in a first instance of the data structure data indicating signatures of a plurality of routines of the interface at a first time, wherein a signature of each routine includes a name of the routine and a type of the routine and parameter types for all parameters of the routine.

6. (Original) A method as recited in Claim 1, wherein:

- each of the interacting computer program modules include instructions causing one or more processors to obtain at least one of a property of a corresponding networking device type of a plurality of networking devices types and an action performed by the corresponding networking device type;
- the first computer program module includes instructions causing one or more processors, based on interacting with a particular interacting computer program module, to perform at least one of communicating with a first device of the corresponding networking device type on a network of networking devices including the first device, and presenting properties of the first device to a network manager for the network, and displaying connections among the networking devices of the network to the network manager; and
- the plurality of networking devices types include one or more models of repeater, a switch, a router, a hub, a bridge, and a gateway.

Art Unit: 2191

7. (Previously Presented) A method as recited in Claim 1, wherein the second version number is assigned when the second module is developed; and compatibility is determined at a later time.

8. (Currently Amended) A computer-implemented method of maintaining version compatibility between a first module and one or more interacting modules that interact with the first module through an interface, the method comprising the steps of:

retrieving data from a stored mapping between a plurality of instances of a data structure describing capabilities of the interface at a corresponding plurality of times and a corresponding plurality of version numbers for the first module;

automatically assigning a second version number for a second module of the one or more interacting modules based on the mapping and based on a description of interface capabilities of the second module;

determining compatibility based on a first version number for the first module and the second version number for the second module;

wherein the step of automatically assigning comprises:

describing a subset of the interface capabilities, which subset is employed by the second module;

determining from the plurality of instances at least one instance including data describing the subset of the interface capabilities; and

assigning the second version number for the second module based on the corresponding version number in the mapping and the at least one instance.

9. (Canceled)

10. (Currently Amended) A method as recited in Claim [[9]] 8, wherein said step of assigning the second version number comprises assigning as the second version number a particular value of the plurality of version numbers for the first computer program module, the particular value

Art Unit: 2191

associated with a particular instance of the at least one instance, the particular instance corresponding to an earliest time of one or more times corresponding to the at least one instance.

11. (Previously Presented) A method as recited in Claim 8, wherein the plurality of version numbers for the first computer program module corresponding to the plurality of instances vary in one direction with time of the plurality of times corresponding to the plurality of instances.

12. (Original) A method as recited in Claim 8, wherein a first instance of the data structure comprises data indicating a routine name and a routine type of a routine of the first module at a first time.

13. (Original) A method as recited in Claim 12, wherein the first data further indicates a parameter type for the routine.

14. (Previously Presented) A method as recited in Claim 8, wherein a first instance of the data structure comprises data indicating signatures of a plurality of routines of the computer program module interface capabilities at a first time, wherein a signature of each routine includes a name of the routine and a type of the routine and parameter types for all parameters of the routine.

15. (Previously Presented) A method as recited in Claim 14, wherein the data indicating signatures of the plurality of routines of the computer program interface capabilities at a first time comprises hashed values, each hashed value uniquely indicating a signature of each routine of the interface.

16. (Original) A method as recited in Claim 14, wherein the plurality of routines comprises all the routines of the interface.

Art Unit: 2191

17. (Original) A method as recited in Claim 14, wherein the plurality of routines comprises all the routines of the interface except routines not implemented in the first module.
18. (Previously Presented) A method as recited in Claim 8, wherein:
the second version number is developed when the second computer program module is developed; and
compatibility is determined at a later time.
19. (Previously Presented) A method as recited in Claim 18, wherein compatibility is determined when the second computer program module is installed for use with the first computer program module.
20. (Previously Presented) A method as recited in Claim 18, wherein compatibility is determined when the second computer program module is invoked for execution by the first module.
21. (Previously Presented) A method as recited in Claim 8, wherein each computer program module of the first computer program module and the one or more interacting computer program modules comprises instructions for causing one or more processors to perform one or more tasks.
22. (Previously Presented) A method as recited in Claim 8, wherein the first computer program module comprises instructions for causing one or more processors to manage a plurality of networking devices in response to data indicating input by a user.
23. (Previously Presented) A method as recited in Claim 8, wherein each interacting computer program module of the one or more interacting computer program modules comprises instructions for causing one or more processors to provide device-specific information for one of a plurality of networking devices.

Art Unit: 2191

24. (Previously Presented) A method as recited in Claim 21, wherein:

each of the interacting computer program modules include instructions causing one or more processors to obtain at least one of a property of a corresponding networking device type of a plurality of networking devices types and an action performed by the corresponding networking device type;

the first module includes instructions causing one or more processors, based on interacting with a particular interacting computer program module, to perform at least one of communicating with a first device of the corresponding networking device type on a network of networking devices including the first device, and presenting properties of the first device to a network manager for the network, and displaying connections among the networking devices of the network to the network manager; and

the plurality of networking devices types include one or more models of repeater, a switch, a router, a hub, a bridge, and a gateway.

25. (Currently Amended) A computer-implemented method of determining version compatibility between a first computer program module and a computer program second module of one or more interacting modules that interact with the first computer program module through an interface, the method comprising the steps of:

obtaining a first version number for the first computer program module;

automatically assigning [obtaining] a second version number for the second computer program module, the second version number set when the second computer program module is developed based on a mapping between a plurality of instances of a data structure describing the interface capabilities at a corresponding plurality of times and a corresponding plurality of version numbers for the first computer program module and based on a description of interface capabilities of the second computer program module; and

determining whether the computer program modules are compatible based on the first version number and the second version number;

wherein the step of automatically assigning comprises:
describing a subset of the interface capabilities, which subset is employed by the second
computer program module;
determining from the plurality of instances at least one instance including data describing
the subset of the interface capabilities; and
assigning the second version number for the second module based on the corresponding
version number in the mapping and the at least one instance.

26. (Previously Presented) A method as recited in Claim 25, wherein:
- each of the interacting computer program modules include instructions causing one or more processors to obtain at least one of a property of a corresponding networking device type of a plurality of networking devices types and an action performed by the corresponding networking device type;
 - the first computer program module includes instructions causing one or more processors, based on interacting with a particular interacting computer program module, to perform at least one of communicating with a first device of the corresponding networking device type on a network of networking devices including the first device, and presenting properties of the first device to a network manager for the network, and displaying connections among the networking devices of the network to the network manager; and
 - the plurality of networking devices types include one or more models of repeater, a switch, a router, a hub, a bridge, and a gateway.
27. (Currently Amended) A computer-readable storage medium for maintaining version compatibility between a first computer program module and one or more interacting computer program modules that interact with the first computer program module through an interface, the computer-readable storage medium carrying:
- a plurality of instances of a data structure describing the interface capabilities at a corresponding plurality of times;

Art Unit: 2191

a mapping that associates the plurality of instances with a corresponding plurality of version numbers for the first computer program module; and
one or more sequences of instructions, which, when executed by one or more processors, cause the one or more processors to carry out the steps of
retrieving data from the mapping, and
automatically assigning [developing] a second version number for a second computer program module of the one or more interacting computer program modules based on the mapping and based on a description of interface capabilities of the second module,
wherein compatibility is determined based on a first version number for the first computer program module and the second version number for the second computer program module;
wherein the step of automatically assigning comprises:
describing a subset of the interface capabilities, which subset is employed by the second computer program module;
determining from the plurality of instances at least one instance including data describing the subset of the interface capabilities; and
assigning the second version number for the second module based on the corresponding version number in the mapping and the at least one instance.

28. (Previously Presented) A system for maintaining version compatibility between a first computer program module and one or more interacting computer program modules that interact with the first computer program module through an interface, the system comprising:
- means for retrieving a stored mapping between a plurality of instances of a data structure describing the interface capabilities at a corresponding plurality of times and a corresponding plurality of version numbers for the first computer program module;
 - means for automatically assigning a second version number for a second computer program module of the one or more interacting computer program modules based

on the mapping and based on a description of interface capabilities of the second module; [[and]]

means for determining compatibility based on a first version number for the first computer program module and the second version number for the computer program second module;

wherein the automatically assigning means comprises:

means for describing a subset of the interface capabilities, which subset is employed by the second computer program module;

means for determining from the plurality of instances at least one instance including data describing the subset of the interface capabilities; and

means for assigning the second version number for the second module based on the corresponding version number in the mapping and the at least one instance.

29. (Currently Amended) A computer system for maintaining version compatibility between a first computer program module and one or more interacting computer program modules that interact with the first computer program module through an interface, the system comprising:

a processor;

a computer-readable storage medium carrying

a stored mapping between a plurality of instances of a data structure describing the interface capabilities at a corresponding plurality of times and a corresponding plurality of version numbers for the first computer program module, and

one or more stored sequences of instructions which, when executed by the processor, cause the processor to carry out the steps of:

retrieving data from the mapping;

automatically assigning a second version number for a second computer program module of the one or more interacting computer program modules based on the mapping and based on a description of interface capabilities of the second module; [[and]]

determining compatibility based on a first version number for the first computer program module and the second version number for the second computer program module;

wherein the step of automatically assigning comprises:

describing a subset of the interface capabilities, which subset is employed by the second computer program module;

determining from the plurality of instances at least one instance including data describing the subset of the interface capabilities; and

assigning the second version number for the second module based on the corresponding version number in the mapping and the at least one instance.

Reasons for allowance

4. The following is an examiner's statement of reasons for allowance: the cited prior art taken alone or in combination fail to disclose, in combination with other claimed limitations, automatically assigning a second version number for a second computer program module of the one or more interacting modules based on a corresponding instance from the plurality of instances of the data structure contained in the mapping and based on second information describing interface capabilities of the second computer program module; wherein the step of automatically assigning comprises: describing a subset of the interface capabilities, which subset is employed by the second computer program module; determining from the plurality of instances at least one instance including data describing the subset of the interface capabilities; and assigning the second version number for the second module based on the corresponding version number in the mapping and the at least one instance as recited substantially in independent claims 1, 8, 25, 27, 28 and 29.

Art Unit: 2191

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

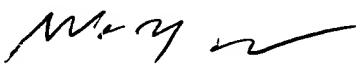
Conclusion

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Wei Y. Zhen whose telephone number is (571) 272-3708. The examiner can normally be reached on Monday-Friday, 8 a.m. - 4:30 p.m..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Wei Zhen
6/7/2005


WEI Y. ZHEN
PRIMARY EXAMINER